← Back **Advanced Segmentation Approaches**
Graded Assignment • 25 min

English **Due** Jan 3, 11:59 PM +06

1. Given two binary masks of puzzle pieces, one that identifies all puzzle pieces ("maskAll") and one that only identifies back-facing puzzle pieces ("maskBack"), how can you combine these masks to obtain only the front-facing puzzle pieces?

1 point

○
```
maskAll & maskBack
```

●
```
maskAll & ~maskBack
```

○
```
~maskAll | maskBack
```

○
```
maskAll | ~maskBack
```

2. Which of the following statement about morphology are true (select all that apply)?

1 point

☑ you must specify the size and shape of a structuring element

☑ you create a structuring element with the `strel` function

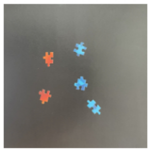
☐ you can only use morphology when improving segmentation of grayscale images

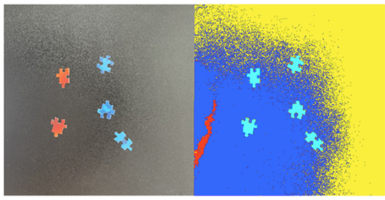
☐ you need to perform spatial filtering before applying morphology

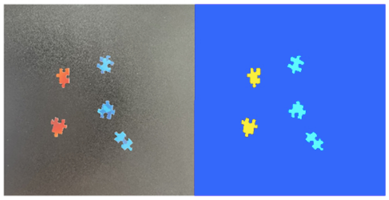
3. Import the image, "Puzzle_06.jpg", found in the course files and convert it to HSV.

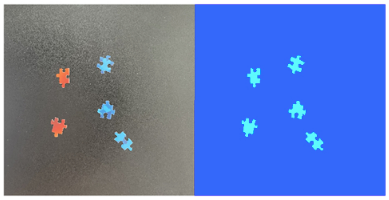
1 point

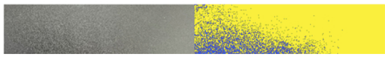
Assume you want to differentiate between the red and blue puzzle pieces. Perform K-means clustering to create a matrix with three labels, one for each color of puzzle piece and the background. Which image below most closely resembles your result?

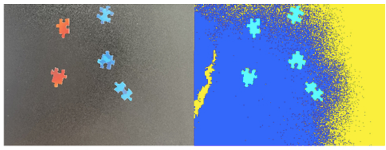


○ 

○ 

○ 

● 



4. Which response below provides the best explanation for the result in the previous question? 1 point

○ Because the number of background pixels is so much larger than puzzle pieces, the `imsegkmeans` does not distinguish between the different colored puzzle pieces.

○ Four clusters should be used for this image: two for the background variation and one for each color.

○ The `imsegkmeans` function returns a labeled matrix that accurately identifies the background and each color of puzzle piece.

● The variation in the background pixels results in the background being divided into multiple labels rather than separating the puzzle pieces by color.

5. Which of the following pieces of code takes a color image, `img`, and uses a binary mask, `BW`, to create a masked image? 1 point

○
```
maskedImage = img;
maskedImage(repmat(BW,1,1,3)) = 0;
```

○
```
maskedImage = img;
maskedImage(~BW) = 0;
```

●
```
maskedImage = img;
maskedImage(repmat(~BW,1,1,3)) = 0;
```

○
```
maskedImage = img;
maskedImage(repmat(~BW,3)) = 0;
```

6. Assume you want to use a rectangular structuring element with size [3,6] to expand then shrink a foreground mask "BW". Which of the following code segments accomplishes this task? 1 point

○
```
se = strel("rectangle")
BW = imdilate(BW, se);
BW = imerode(BW, se);
```

○
```
BW = imdilate(BW, "rectangle", [3,6]);
BW = imerode(BW, "rectangle", [3,6]);
```

●
```
se = strel("rectangle",[3,6]);
BW = imdilate(BW, se);
BW = imerode(BW, se);
```

○
```
se = strel("rectangle",[3,6]);
BW = imopen(BW, se);
BW = imclose(BW, se);
```